

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

**Методичні вказівки до виконання курсової роботи
по курсу «Алгоритми та структури даних»**

для студентів, які навчаються за напрямком
6.050103 «Програмна інженерія» спеціальності .05010301 «Програмне
забезпечення систем»

Затверджено
редакційно-видавничою
радою НТУ «ХПІ», протокол № 2
від «23» червня 2016 р.

Харків
НТУ «ХПІ»
2016

Методичні вказівки до виконання курсової роботи по курсу «Алгоритми та структури даних» для студентів, які навчаються за напрямком 6.050103 «Програмна інженерія» спеціальності .05010301 «Програмне забезпечення систем» / укл. Н.К. Стратієнко, О.В. Шматко, І.О. Бородіна. – Х. : НТУ «ХП», 2016. – 34 с.

Укладачі: Н.К. Стратієнко
О.В. Шматко
І.О. Бородіна

Рецензент В.О. Гужва

Кафедра програмної інженерії та інформаційних технологій управління

ЗМІСТ

Вступ.....	5
1 Мета і задачі виконання курсової роботи.....	6
2 Основні етапи виконання курсової роботи	7
2.1 Постановка задачі дослідження.....	8
2.2 Побудова математичної моделі	8
2.3 Вивчення теоретичних основ використовуваних методів	9
2.4 Розробка та тестування алгоритмічного та програмного забезпечення ..	9
3 Зміст курсової роботи	10
3.1 Вступ	10
3.2 Розділ 1 Теоретичні основи розробки та аналізу алгоритмів. Постановка задачі.....	10
3.3 Розділ 2 Теоретичні основи розробки програмного забезпечення	11
3.4 Розділ 3 Опис розробленого програмного забезпечення	12
3.5 Розділ 4 Використання розробленого програмного забезпечення	13
3.6 Висновки	15
3.7 Список джерел інформації	15
4 Вимоги до програмного забезпечення, що розробляється	16
4.1 Вибір засобів розробки прикладного програмного забезпечення	16
4.2 Структура прикладного програмного забезпечення.....	17
4.2.1 Введення, редагування і збереження початкових даних	17
4.2.2 Вирішення основної задачі	17
4.2.3 Перевірка отриманого рішення.....	18
4.2.4 Збереження звіту про виконані дослідження	18
4.2.5 Отримання довідки.....	18
5 Захист курсової роботи	19
5.1 Загальний порядок захисту курсової роботи	19
5.2 Вимоги до презентаційних матеріалів.....	19
5.3 Вимоги до доповіді.....	20
5.4 Вимоги до демонстрації програмного забезпечення	20
6 Критерії оцінювання курсової роботи.....	21
7 Варіанти завдань на курсову роботу	22
7.1 Дослідження ефективності квадратичних та оптимальних алгоритмів сортування	22
7.2 Реалізація та аналіз якості генераторів випадкових чисел.....	22

7.3 Візуалізація базових алгоритмів пошуку на графах	22
7.4 Побудова кістяка для графа транспортних маршрутів	23
7.5 Визначення порядку збирання програмного комплексу з урахуванням залежностей між компонентами	23
7.6 Побудова опуклої оболонки на площині	24
7.7 Пошук найкоротшого шляху в двовимірному лабіринті	24
7.8 Реалізація шифрованого обміну повідомленнями	24
7.9 Наближене рішення задачі комівояжера	25
7.10 Кластеризація живих організмів на підставі ступеня близькості їхньої ДНК	25
7.11 Розробка прикладного застосування для архівації даних на базі кодів Хаффмана	25
Список літератури	26

ВСТУП

Методичні вказівки до виконання курсової роботи по курсу «Алгоритми та структури даних» є частиною загального методичного забезпечення курсу «Алгоритми та структури даних».

Курсова робота виконується в 3-му семестрі.

Виконання курсової роботи повинне забезпечити закріплення теоретичних знань і практичних навичок, отриманих при вивченні лекційної та практичної частин курсу.

У методичних вказівках розглянуті основні питання, пов'язані з виконанням курсової роботи, оформленням пояснювальної записки до курсової роботи, захистом курсової роботи.

1 МЕТА І ЗАДАЧІ ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Курсова робота виконується з метою закріплення знань по курсу «Алгоритми та структури даних» і отримання навичок розробки програмного забезпечення, що вимагає застосовувати вивчені структури даних та алгоритми.

Завданнями курсової роботи є:

- набуття навичок у розробці програм, які включають нетривіальні алгоритми;
- розробка алгоритмічного забезпечення;
- розробка програмного забезпечення.

Результати виконання курсової роботи надаються у вигляді розробленого програмного забезпечення, пояснювальної записки до курсової роботи, презентаційного матеріалу.

Пояснювальна записка до курсової роботи – основний документ, який відбиває усі етапи і результати виконаної роботи. Оформлення пояснювальної записки повинне виконуватися відповідно до норм, які розписані в нормативно-методичному виданні НТУ «ХПІ» «Текстові документи у сфері учбового процесу. Основні вимоги до виконання».

Пояснювальна записка повинна містити:

- завдання на курсову роботу;
- відгук керівника;
- реферати українською, російською та англійською мовами;
- зміст;
- вступ;
- основну частину;
- висновок;
- список джерел інформації;
- додатки.

2 ОСНОВНІ ЕТАПИ ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Курсова робота по курсу «Алгоритми та структури даних» передбачає розробку алгоритмічного і програмного забезпечення згідно отриманому завданню.

Виконання курсової роботи розбивається на наступні етапи:

- постановка задачі дослідження;
- побудова математичної моделі;
- вивчення теоретичних основ алгоритмів та поглиблене вивчення структур даних, які будуть використовуватись;
- розробка та тестування алгоритмічного та програмного забезпечення;
- вирішення задачі та аналіз результатів;
- оформлення пояснювальної записки та презентації;
- захист курсової роботи.

Усі етапи необхідно виконувати в терміни, вказані в графіку виконання. Календарний план виконання курсової роботи представлений у табл. 2.1.

Таблиця 2.1 – Календарний план виконання курсової роботи по курсу «Алгоритми та структури даних»

№ з/п	Найменування етапу виконання к/р	Терміни виконання
1	Вивчення теоретичних основ використаних математичних методів. Постановка задачі дослідження.	27 вересня
2	Алгоритмічне забезпечення рішення задачі дослідження	15 жовтня
3	Розробка структури програмного забезпечення	30 жовтня
4	Розробка програмного забезпечення. Програмна реалізація математичних методів, які використовуються	
5	Рішення задачі та аналіз отриманих результатів	10 листопада
6	Оформлення пояснювальної записки	17 грудня
7	Підготовка презентації курсової роботи	22 грудня
8	Захист курсової роботи	25 грудня

Дослідження підрозділяється на три етапи.

На *підготовчому етапі* дослідження кожен студент повинен:

- ознайомитись зі змістовною постановкою отриманої задачі;
- вивчити алгоритми та структури даних, необхідні, щоб виконати завдання;
- розробити алгоритмічне та програмне забезпечення;
- побудувати набори тестових даних.

На *робочому етапі* дослідження студенту необхідно:

- для отриманих тестових даних на основі розробленого програмного забезпечення вирішити поставлену задачу, щоб перевірити правильність побудованих алгоритмів;
- вирішити поставлену задачу для даних великої розмірності, щоб визначити межі використання програмного забезпечення.

На *завершальному етапі* дослідження студент повинен:

- інтерпретувати отримані результати в термінах предметної області змістовної постановки задачі;
- зробити висновки по результатам виконання роботи.

2.1 Постановка задачі дослідження

Постановка задачі дослідження – найбільш відповідальний етап операційного дослідження, тому що помилки на цьому етапі можуть зробити марними матеріальні і трудові витрати на усіх наступних етапах. Спочатку задачу формують з точки зору замовника. Після з'ясування мети проводиться ретельне обстеження об'єкту, вивчається множина чинників, що впливають на протікання процесів. Після збору даних обстеження і їх аналізу дослідник виділяє сукупність істотних чинників, проводить консультації із замовником і уточнює остаточно якісну (змістовну) постановку основної задачі.

Для курсової роботи ситуація спрощена: потрібно користуватися формулюванням із завдання, але за додатковою інформацією можливо звертатися до керівника.

2.2 Побудова математичної моделі

Маючи строгу, логічно несуперечливу змістовну постановку задачі, студент здійснює її формалізацію шляхом побудови математичної моделі, що містить відповідні критерій оптимальності та обмеження. Апарат для побудови такої моделі вивчається у лекційному курсі.

2.3 Вивчення теоретичних основ використовуваних методів

Маючи математичну модель, студент здійснює класифікацію задачі та алгоритмів, що потрібно застосувати; виконує огляд основних алгоритмів цього класу; детально вивчає алгоритми та структури даних, вказані в завданні.

2.4 Розробка та тестування алгоритмічного та програмного забезпечення

Програмне забезпечення має бути дружнім до дослідника та повинне максимальним чином використовувати принципи сучасних інформаційних технологій.

Вимоги до програмного забезпечення наводяться на діаграмі варіантів використання.

Багато варіантів завдань в цій курсовій роботі передбачають покрокове виконання та графічну інтерпретацію. Додатково потрібно згенерувати звіт за результатами роботи програми у форматі plain text, HTML, LaTeX або PDF. Крім того, у будь-який момент користувач повинен мати можливість отримати довідку про роботу програми.

Відповідно до завдання на курсову роботу, головне вікно програми повинне включати, окрім меню, ще й можливості для перегляду і редагування даних, кнопки для виконання деяких функцій, а також елементи відображення результатів. Доцільно розмістити панелі для графічного відображення результатів.

Багато варіантів завдань потребують читання даних із великого бінарного файлу. Рекомендується також додати можливість зберігати, а також читати дані у текстовому форматі.

Окремо слід створити невелике діалогове вікно, яке містить назву програми і інформацію про автора.

Тестування програмного забезпечення проводиться на задачах невеликої розмірності, які можна вирішити вручну і порівняти з розрахунками програми.

3 ЗМІСТ КУРСОВОЇ РОБОТИ

У цьому розділі розглядається зразкова структура записки по курсовій роботі. Записка складається з вступу, чотирьох розділів, висновків і додатків. У додатки можна помістити ілюстративний матеріал, таблиці, формули. Мінімальний об'єм записки – 40 сторінок.

До записки по курсовій роботі може додаватися носій, що містить повну інформацію по матеріалах, розроблених в результаті виконання курсової роботи:

- вихідні коди розробленого програмного забезпечення;
- виконувані файли прикладного програмного забезпечення, наприклад, форматів ELF або JAR, додавати Portable Executable (.exe) не рекомендується;
- повний текст записки.

Дамо коротку характеристику змісту розділів і підрозділів записки. Варіант, що наводиться, є рекомендаційним. Він відбиває той необхідний об'єм інформації, який має бути наведений в записці. При виконанні курсової роботи склад і зміст розділів записки можуть бути змінені студентом за узгодженням з керівником курсової роботи.

3.1 Вступ

У вступі розглядаються цілі, які мають бути досягнуті в результаті виконання курсової роботи, обґрунтовується актуальність теми курсової роботи. Приблизний об'єм вступу – 1-1,5 сторінок.

3.2 Розділ 1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ТА АНАЛІЗУ АЛГОРИТМІВ. ПОСТАНОВКА ЗАДАЧІ

Основна мета цього розділу – показати, що студент ознайомився з основами теорії алгоритмів та необхідними структурами даних, і сформулювати основні вимоги до програмного забезпечення, що розробляється.

Підрозділ 1.1 Математичні основи аналізу алгоритмів

У підрозділі необхідно дати визначення таким основним поняттям теорії алгоритмів, як складність алгоритму, алгоритми «розділяй-та-володарюй», динамічне програмування, амортизаційний аналіз та ін.

Приблизний об'єм підрозділу – 3-4 сторінки.

Підрозділ 1.2 Структури даних і класифікація основних алгоритмів

У підрозділі треба дати характеристику структурам даних і класифікацію основних алгоритмів: алгоритми на графах, динамічне програмування, геометричні алгоритми тощо.

Приблизний об'єм підрозділу – 3-4 сторінки

Підрозділ 1.3 Постановка задачі

У підрозділі необхідно:

- навести математичну моделі задачі;
- привести якісну постановку задачі;
- описати основні цілі, які потрібно досягти в результаті виконання курсової роботи.

До основних задач курсової роботи відносяться:

- вибір платформи (операційної системи) для програмного забезпечення;
- вибір засобів розробки прикладного програмного забезпечення;
- розробка прикладного програмного забезпечення;
- налагоджування та тестування прикладного програмного забезпечення;
- дослідна експлуатація прикладного програмного забезпечення на контрольних даних і аналіз отриманих результатів.

Приблизний об'єм розділу – 6-8 сторінок.

3.3 Розділ 2 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розділ присвячується математичному забезпеченню вирішення задачі.

Розділ 2.1 Структури даних і класифікація основних алгоритмів заданого класу

У підрозділі слід охарактеризувати заданий клас алгоритмів та структури даних, наприклад, описати алгоритми на графах та способи подання графу в пам'яті.

Приблизний об'єм підрозділу – 2-3 сторінки.

Розділ 2.2 Опис використаних алгоритмів

У підрозділі необхідно описати конкретні алгоритми, які буде використано в роботі. Слід навести схеми алгоритмів у вигляді добре структуровано-

го псевдокоду або справжнього коду. Показати, як визначається складність описаних алгоритмів та навести приклади роботи алгоритму на простих даних.

Приблизний об'єм підрозділу – 6-9 сторінок.

3.4 Розділ 3 ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Основна мета цього розділу – описати результати, отримані в процесі розробки інформаційного і програмного забезпечення, що розробляється для вирішення прикладних задач.

Підрозділ 3.1 Визначення варіантів використання програмного забезпечення

У підрозділі необхідно визначити основні типи користувачів, що працюють з прикладним програмним забезпеченням, сформулювати і описати варіанти використання. Необхідно привести діаграму варіантів використання.

Приклад можливої побудови діаграми варіантів використання приведений на рисунку 3.1.

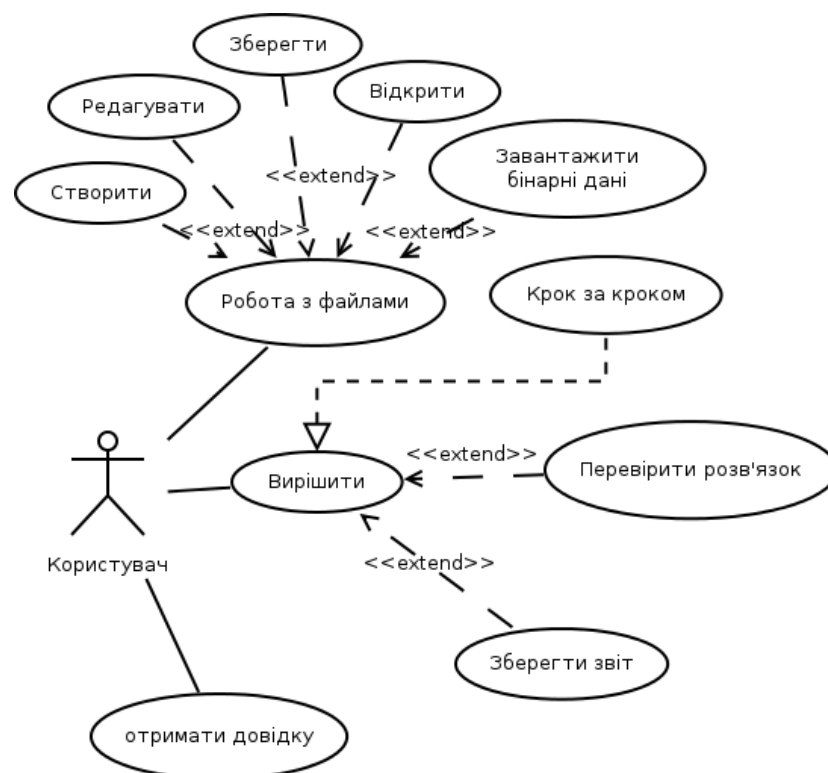


Рисунок 3.1 – Діаграма варіантів використання

Приблизний об'єм підрозділу – 2-3 сторінки.

Підрозділ 3.2 Обґрунтування вибору операційної системи і засобів розробки прикладного програмного забезпечення

У цьому підрозділі мають бути розглянуті можливості 2-3 операційних систем та 2-3 засобів розробки прикладного програмного забезпечення. Мають бути приведені основні переваги і недоліки даних програмних продуктів.

Підрозділ повинен закінчуватися обґрунтуванням вибору операційної системи та засобів розробки прикладного програмного забезпечення.

Приблизний об'єм підрозділу – 3-6 сторінок.

Підрозділ 3.3 Структура застосування

У цьому підрозділі мають бути описані основні елементи прикладного програмного забезпечення і взаємозв'язку між ними. Необхідно представити UML-діаграму компонентів і діаграму класів і описати їх словами.

Діаграма компонентів показує різні компоненти системи і зв'язки між ними. Компонент є фізичним модулем програмного коду. Компонент часто вважають синонімом пакету, але ці поняття можуть відрізнятися, оскільки компоненти є фізичним об'єднанням програмного коду. Хоча окремий клас може бути представлений в цілій сукупності компонентів, цей клас має бути визначений тільки в одному пакеті. Наприклад, клас String в мові Java є частиною пакету java.lang, але він може міститися у ряді компонентів.

Залежності між компонентами показують, як зміни одного компонента можуть вплинути на зміни інших компонентів. Існує досить обмежена кількість видів залежностей, які можна використовувати, включаючи залежності типу «зв'язок» і «компіляція».

Діаграма класів описує типи об'єктів системи і різного роду статичні стосунки, які існують між ними. Є два основні види статичних відносин:

- асоціації (наприклад, клієнт може узяти напрокат ряд відеокaset);
- підтипи (медсестра є різновидом особи).

На діаграмах класів зображуються також атрибути класів, операції класів і обмеження, які накладаються на зв'язки між об'єктами.

Приблизний об'єм підрозділу – 2-5 сторінок.

3.5 Розділ 4 Використання розробленого програмного забезпечення

Основна мета цього розділу – описати результати, отримані в процесі використання розробленого інформаційного і програмного забезпечення для вирішення прикладних завдань. У цьому розділі описується послідовність роботи користувача з прикладним програмним забезпеченням, а також опи-

суються і аналізуються результати, отримані при вирішенні практичних завдань.

Підрозділ 4.1 Установка програмного забезпечення

У цьому підрозділі описується процедура установки розробленого прикладного програмного забезпечення на комп'ютер користувача. Процедuru установки рекомендується описати у вигляді послідовності кроків, що виконуються користувачем в процесі установки. Рекомендується також привести основні вимоги до апаратного і програмного забезпечення: тип операційної системи, наявність спеціального програмного забезпечення, тип процесора, мінімальний об'єм оперативної пам'яті, наявність вільного місця на жорсткому диску і так далі.

Якщо обрано операційну систему GNU/Linux, рекомендовано виконати пакування програмного забезпечення (rpm, deb, або ebuild) – це значно спрощує установку залежностей програми та її видалення із системи.

Необхідно описати послідовність дій при першому запуску прикладного програмного забезпечення (тобто як запустити програму і як з неї вийти).

Приблизний об'єм підрозділу – 1-2 сторінки.

Підрозділ 4.2 Інструкція користувачеві

У цьому підрозділі приводиться опис роботи користувача з розробленим прикладним програмним забезпеченням.

Цей опис можна розглядати як фрагмент посібника користувача, що входить в комплект документації на програмний продукт. Опис має бути досить детальним, дозволяти працювати з прикладним програмним забезпеченням користувачеві, що не має спеціальної підготовки. Опис повинен ілюструватися прикладами екранних форм, які розташовуються в тексті у вигляді малюнків. Приклад головної екранної форми наведений на рисунку 3.2.

Приблизний об'єм підрозділу – 6-10 сторінок.

Підрозділ 4.3 Аналіз результатів

У цьому підрозділі приводиться опис результатів, отриманих при використанні прикладного програмного забезпечення.

Потрібно навести інтерпретацію фрагменту або всіх вхідних даних великої розмірності у вигляді таблиці чи рисунків. Можна навести відповідні скріншоти екранних форм.

Потрібно навести результат та проаналізувати його.

Приблизний об'єм підрозділу – 2-4 сторінки.

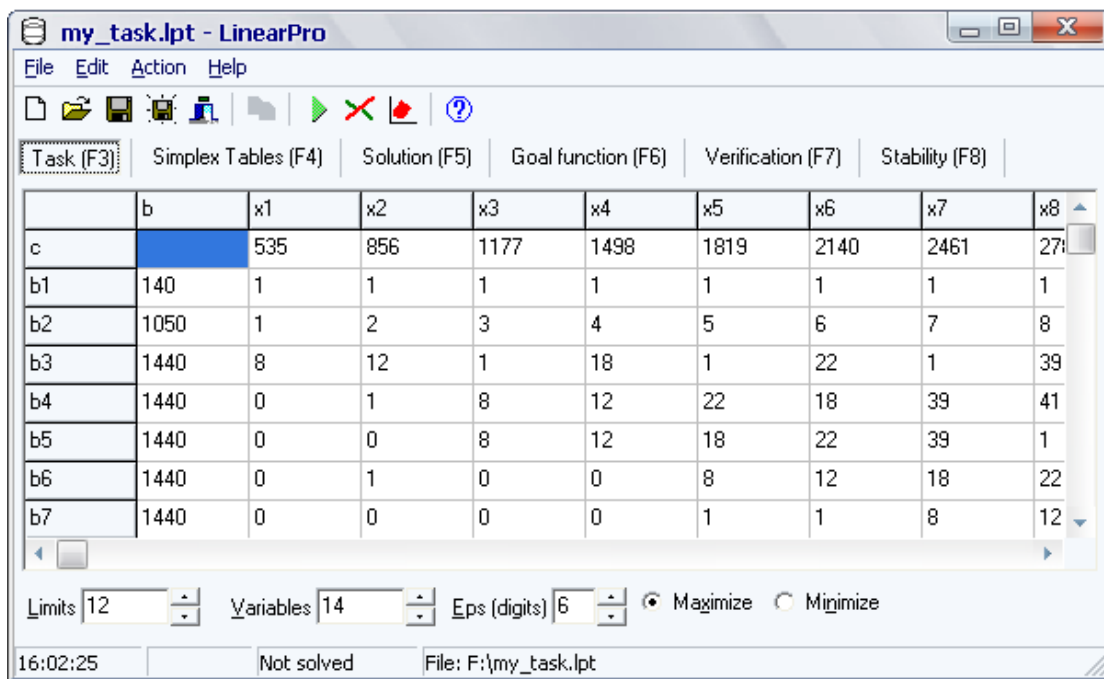


Рисунок 3.2 – Вигляд головної екранної форми застосування

3.6 Висновки

У висновках підводяться підсумки виконання курсової роботи, коротко перераховуються отримані результати, робляться висновки про працездатність розробленого застосування і доцільність його використання для вирішення практичних завдань.

Об'єм висновків – 1 сторінка.

3.7 Список джерел інформації

У розділі наводиться перелік цитованих, розглянутих, згаданих і використаних джерел інформації. Джерелами інформації є книги, статті, нормативно-технічні документи, дисертації і так далі. Джерела інформації записують без групувань на ті, що мають та не мають посилань у тексті.

Джерела, на які є посилання в тексті, розміщують у порядку їхньої появи.

Джерела, на які нема посилань в тексті, розміщують наступним чином:

- 1) джерела під редакцією одного із співавторів, їх записують в алфавітному порядку заголовків;
- 2) книги, статті, рецензії, реферати, їх записують в алфавітному порядку прізвищ авторів;
- 3) нормативно-технічні документи, що групуються за категоріями, а усередині категорій – за збільшенням реєстраційних номерів.

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ЩО РОЗРОБЛЯЄТЬСЯ

4.1 Вибір засобів розробки прикладного програмного забезпечення

Вибір засобів розробки прикладного програмного забезпечення необхідно виконувати з урахуванням наступних чинників:

- наявність досвіду роботи з цим програмним продуктом;
- доступність цього програмного продукту, зокрема, наявність ліцензій;
- ефективність його використання при розробці прикладного програмного забезпечення;
- швидке візуальне проектування компонентів прикладного програмного забезпечення;
- зручність налагодження;
- можливість отримання консультацій.

Завдання можна виконувати мовою програмування на вибір:

- C;
- C++;
- Pascal (зараз мало використовується);
- Java;
- C#;
- Python;
- Ruby.

Не рекомендовано використовувати мову асемблера, яка додає до алгоритму забагато деталей.

Програму можна демонструвати у будь-якій операційній системі. Рекомендовано розробляти застосування так, щоб його можна було зібрати та виконувати на широкому класі систем (це елементарно досягти на перелічених мовах). Рекомендовано використовувати будь-який дистрибутив GNU/Linux або FreeBSD.

Перелічимо кілька важливих вільних або безкоштовних редакторів та середовищ розробки:

- Vim;
- Emacs;
- Eclipse;

- NetBeans;
- IntelliJ IDEA;
- KDevelop;
- MonoDevelop;
- Lazarus (вільне середовище, подібне до Delphi).

4.2 Структура прикладного програмного забезпечення

Застосування, що розробляється, повинне реалізовувати наступні функції:

- введення, редагування і збереження вихідних даних;
- вирішення основної задачі;
- перевірка отриманого рішення;
- перегляд пошуку рішення покроково;
- збереження звіту про виконані дослідження;
- отримання довідки.

У наступних підрозділах описані склад і особливості кожної програми.

4.2.1 Введення, редагування і збереження початкових даних

Необхідно забезпечити введення початкових даних вручну або з файлу, редагування даних і їх збереження у файл. Формат файлу довільний. Рекомендовано використовувати текстовий формат (plain text, XML, JSON).

Багато варіантів завдань потребують читання даних із великого бінарного файлу. Збереження у бінарному форматі можна не реалізовувати, але збереження до базового формату (текстового) потрібно підтримувати.

4.2.2 Вирішення основної задачі

Необхідно забезпечити рішення задачі заданими алгоритмами. У разі несумісних умов або некоректного формату вхідних даних застосування повинне повідомити про це користувача. Програма повинна функціонувати так, що для будь-якого входу вона або видає розв'язок, або повідомляє про помилку.

Крім того, застосування повинне надавати можливість перегляду процесу пошуку розв'язку по кроках, навіть якщо задача має несумісні умови, що дозволяють виконати кілька кроків алгоритму.

4.2.3 Перевірка отриманого рішення

Багато варіантів курсової роботи допускають перевірку отриманого рішення.

1 Перевірка може бути виконана користувачем за допомоги графічної інтерпретації, наприклад, елементи відсортованої послідовності можна зобразити у вигляді гістограми.

2 Такі завдання, як генерація випадкових послідовностей, включають формальні тести якості отриманого результату.

4.2.4 Збереження звіту про виконані дослідження

Програма повинна забезпечувати збереження звітів про отримане рішення. Рекомендуються формати plain text, HTML, LaTeX (із компіляцією до PDF) або PDF.

У звіт про отримане рішення необхідно вивести:

- умову задачі;
- кроки алгоритму;
- розв'язок;
- графічну або табличну інтерпретацію результату.

4.2.5 Отримання довідки

Користувач повинен мати можливість ознайомитися з інформацією, достатньою для використання можливостей прикладного програмного забезпечення і режимами їх реалізації. Формат довідки довільний. Рекомендовано застосовувати HTML або PDF. Можливо написання довідки у форматах reStructured Text або LaTeX із наступною компіляцією до HTML або PDF.

5 ЗАХИСТ КУРСОВОЇ РОБОТИ

5.1 Загальний порядок захисту курсової роботи

До захисту курсової роботи допускаються студенти, що виконали курсову роботу в повному обсязі, про що свідчить записка по курсовій роботі, підписана керівником курсової роботи. Оформлення записки повинне відповідати вимогам НТУ «ХПІ». Головні висновки керівника про відповідність змісту курсової роботи завданню, міри самостійності виконання курсової роботи і т.д. мають бути відбиті у відгуку керівника.

Крім пояснювальної записки студент повинен представити презентаційні матеріали і прикладне програмне забезпечення, розроблене в процесі виконання курсової роботи.

Захист курсової роботи проходить з обов'язковим використанням комп'ютерної техніки. При підготовці до захисту студенту необхідно завчасно встановити на наданому йому комп'ютері презентаційні матеріали і прикладне програмне забезпечення. Захист курсової роботи починається з доповіді студента, після закінчення якої він повинен відповісти на поставлені йому питання з теми курсової роботи.

Захист курсової роботи є публічним, тобто на захисті можуть бути присутніми усі, хто бажає, і ставити будь-які питання з теми курсової роботи.

5.2 Вимоги до презентаційних матеріалів

Матеріали, що використовуються як презентаційні, мають бути присутні в пояснювальній записці до курсової роботи у вигляді рисунків, таблиць, схем і тому подібне, розміщених в тексті. Якщо презентаційні матеріали в тексті записки відсутні (наприклад, у зв'язку з тим, що в цих матеріалах представлена інформація з різних розділів записки до курсової роботи), то вони мають бути приведені в додатках до записки по курсовій роботі.

Зразковий перелік презентаційних матеріалів такий:

- 1 Постановка задачі дослідження.
- 2 Класифікація основних алгоритмів.
- 3 Опис клас представлених алгоритмів.
- 4 Схема роботи реалізованих алгоритмів.
- 5 Діаграма варіантів використання.
- 6 Вимоги до програмного забезпечення.
- 7 Середовище розробки.

8 Введення даних.

9 Результати: графічна або табличну інтерпретація.

10 Результати: перевірка рішення.

11 Висновки.

Презентаційні матеріали можуть бути виконані в паперовому або електронному вигляді. Паперові презентаційні матеріали виконуються на папері формату А4 в машинописному вигляді (тобто мають бути надруковані на принтері). Усі написи і рисунки мають бути чіткими, добре читатися. Рукописний варіант презентаційних матеріалів не допускається. Електронні презентаційні матеріали виконуються з використанням відповідного програмного забезпечення (LibreOffice Impress, Microsoft Power Point і тому подібне) і демонструються за допомогою комп'ютера.

5.3 Вимоги до доповіді

Мета доповіді – викласти цілі курсової роботи, виділити і охарактеризувати основні етапи її виконання і отримані результати. Час доповіді – до 5 хвилин. Під час доповіді студент повинен користуватися презентаційними матеріалами. Після завершення доповіді студент переходить до демонстрації розробленого програмного забезпечення.

5.4 Вимоги до демонстрації програмного забезпечення

Основне завдання демонстрації програмного забезпечення – показати працездатність розробленого програмного забезпечення, його основні функціональні можливості, зручність роботи користувача і т.д.

Під час демонстрації студент повинен показати, як працювати з програмним забезпеченням в режимі:

- введення і модифікації початкових даних;
- побудови розв'язку задачі;
- демонстрації та перевірки розв'язку.

6 КРИТЕРІЇ ОЦІНЮВАННЯ КУРСОВОЇ РОБОТИ

На оцінку курсової роботи впливають такі фактори:

1 Помилки, збої тощо у роботі прикладного програмного забезпечення, виявлені при його демонстрації в процесі захисту курсової роботи.

2 Неякісна підготовка доповіді студентом.

3 Неякісні презентаційні матеріали, які не відбивають повною мірою особливості предметної області, результати, отримані при виконанні курсової роботи тощо.

При оцінюванні роботи також ураховується відгук наукового керівника, в якому повинні бути відбиті такі дані:

- актуальність теми виконуваної роботи;
- ступінь самостійності виконання роботи студентом;
- основні результати, отримані при виконанні роботи;
- оцінка роботи за національною шкалою і шкалою ECTS.

7 ВАРІАНТИ ЗАВДАНЬ НА КУРСОВУ РОБОТУ

7.1 Дослідження ефективності квадратичних та оптимальних алгоритмів сортування

Вхідні дані: масив з N цілих чисел, $N = 10$ для тестів і $N = 10\,000\,000$ для демонстрації.

Джерело даних: обраний довільно відеофайл, дані якого обробляються як послідовність 32-бітних чисел у форматі Big Endian.

Вимоги до інтерфейсу: користувач повинен мати можливість переглянути кожних крок виконання будь-якого алгоритму сортування та отримати статистичні відомості: скільки перестановок елементів довелося виконати, скільки мілісекунд знадобилося для сортування тощо.

Використовувані алгоритми: сортування вставками, бульбашкою, швидко, злиттям, купою (один квадратичний та один оптимальний).

7.2 Реалізація та аналіз якості генераторів випадкових чисел

Вхідні дані: довжина генерується послідовності N і початкові параметри, якщо вони потрібні генератору; $N = 100$ для тестів і $N = 10\,000$ для демонстрації.

Вимоги до інтерфейсу: користувач повинен мати можливість переглянути всю згенеровану послідовність, поспостерігати її графічні інтерпретації:

- гістограма розподілу елементів послідовності;
- розподіл на площині (елементи попарно обробляються як координати точок (x, y));
- автокореляція (користувач задає зсув для копії послідовності);

Також потрібно виконати перевірку згідно будь-яким (на вибір студента) чотирьом тестам з пакету NIST.

Використовувані алгоритми: лінійний конгруентний метод чи метод Фібоначчі із затримуванням, а також: генератор із стандартної бібліотеки мови програмування, читання з файлів `/dev/random`, `/dev/urandom` та довільного іншого файлу.

7.3 Візуалізація базових алгоритмів пошуку на графах

Вхідні дані: граф з кількістю вершин N , заданий списком своїх M ребер; $N, M = 10$ для тестів та $N = 256, M = 2048$ для демонстрації.

Джерело даних: обраний довільно відеофайл, дані якого обробляються як послідовність пар однобайтових чисел – номерів вершин, поєднаних ребром.

Вимоги до інтерфейсу: користувач повинен мати можливість переглянути граф на площині та переміщувати його вершини за допомогою миші для кращого вигляду; результати переміщення, так само як і сам граф, можна зберегти до файлу у форматі, обраному студентом (можна базуватися на plain text, XML, JSON); користувач повинен мати можливість обрати вершину та запустити процес пошуку із неї, поряд з кожною вершиною повинні виводитися її статус (біла, сіра, чорна) та час приходу та уходу.

Використовувані алгоритми: пошук в глибину та пошук в ширину.

7.4 Побудова кістяка для графа транспортних маршрутів

Вхідні дані: граф з кількістю вершин N , заданий списком своїх M ребер; $N, M = 10$ для тестів та $N = 256, M = 2048$ для демонстрації.

Джерело даних: обраний довільно відеофайл, дані якого обробляються як послідовність пар однобайтових чисел – номерів вершин, поєднаних ребром.

Вимоги до інтерфейсу: користувач повинен мати можливість переглянути граф на площині та переміщувати його вершини за допомогою миші для кращого вигляду; результати переміщення, так само як і сам граф, можна зберегти до файлу у форматі, обраному студентом (можна базуватися на plain text, XML, JSON); користувач повинен мати можливість переглянути процес побудови кістяка покроково та отримати результат.

Використовувані алгоритми: алгоритм Прима чи алгоритм Крускала.

7.5 Визначення порядку збирання програмного комплексу з урахуванням залежностей між компонентами

Вхідні дані: направлений ациклічний граф (DAG).

Джерело вхідних даних: відомості про залежність програм на комп'ютері користувача, наприклад, з файлу `/var/lib/dpkg/available`.

Вимоги до інтерфейсу: користувач повинен мати можливість переглянути граф на площині та переміщувати його вершини за допомогою миші для кращого вигляду; користувач повинен мати можливість переглянути процес побудови порядку збирання програм та зберегти результати до файлу.

Використовувані алгоритми: топологічне сортування.

7.6 Побудова опуклої оболонки на площині

Вхідні дані: набір з N точок, заданий своїми цілочисельними координатами; $N = 20$ для тестів та $N = 256$ для демонстрації.

Джерело даних: обраний довільно відеофайл, дані якого обробляються як послідовність пар однобайтових чисел – координат вершин.

Вимоги до інтерфейсу: користувач повинен мати можливість переглянути точки на площині, переміщувати їх за допомогою миші; результати переміщення можна зберегти до файлу у форматі, обраному студентом (можна базуватися на plain text, XML, JSON); користувач повинен мати можливість переглянути процес побудови покроково; користувач повинен мати можливість вказати за допомогою миші довільну точку на площині, щоб програма повідомила, чи належить вона до багатокутника, що відповідає оболонці.

Використовувані алгоритми: алгоритм Джарвіса чи алгоритм Грехема.

7.7 Пошук найкоротшого шляху в двовимірному лабіринті

Вхідні дані: набір з N стін, заданий двома парами цілочисельних координат своїх вершин; $N = 20$ для тестів та $N = 50$ для демонстрації.

Джерело даних: обраний довільно відеофайл, дані якого обробляються як послідовність однобайтових чисел – координат вершин.

Вимоги до інтерфейсу: користувач повинен мати можливість переглянути стіни на площині, переміщувати їх за допомогою миші; користувач має можливість обрати дві точки, щоб програма побудувала між ними найкоротший шлях, що не перетинає жодної стіни, або повідомила, що такого шляху не існує; має бути можливість переглянути етапи побудови маршруту.

Використовувані алгоритми: алгоритм Дейкстри чи алгоритм Флойда-Воршола.

7.8 Реалізація шифрованого обміну повідомленнями

Вхідні дані: текстовий файл, що обирає користувач, параметри алгоритму.

Вимоги до інтерфейсу: користувач має можливість запустити дві копії програми на різних комп'ютерах, щоб вони обмінювалися даними через мережу, або зберегти зашифроване повідомлення до файлу з можливістю розшифрувати його.

Використовувані алгоритми: RSA або шифр Віжинера.

7.9 Наближене рішення задачі комівояжера

Вхідні дані: N міст на площині, заданих своїми цілочисленними координатами; $N = 20$ для тестів та $N = 256$ для демонстрації.

Джерело даних: обраний довільно відеофайл, дані якого обробляються як послідовність одnobайтових чисел — координат вершин.

Вимоги до інтерфейсу: користувач повинен мати можливість переглянути міста на площині, переміщувати їх за допомогою миші; користувач має можливість переглянути етапи побудови маршруту комівояжера та результат із вказанням довжини маршруту та його послідовності на площині.

Використовувані алгоритми: наближене рішення задачі комівояжера.

7.10 Кластеризація живих організмів на підставі ступеня близькості їхньої ДНК

Вхідні дані: N зразків ДНК довжиною M нуклеотидів різноманітних живих організмів; $N, M = 20$ для тестів та $N, M = 256$ для демонстрації.

Джерело даних: обраний довільно відеофайл, дані якого обробляються як послідовність пар бітів, кожна з яких кодує один нуклеотид (аденін, тимін, гуанін, цитозин).

Вимоги до інтерфейсу: користувач повинен мати можливість обрати пару зразків ДНК та отримати їхню найбільшу спільну підпослідовність, а також побудувати дендрограму (кладограму) для всіх розглянутих організмів.

Використовувані алгоритми: пошук найбільшої спільної підпослідовності, агломеративний чи дивізімний алгоритм ієрархічної кластеризації.

7.11 Розробка прикладного застосування для архівації даних на базі кодів Хаффмана

Вхідні дані: файл, що обирає користувач, довжина блоку.

Вимоги до інтерфейсу: користувач має можливість виконати стискання даних та розпакувати архів, переглянути статистику густоти кожного символу, додатково вказати довжину блоків, на які розділяється файл при стисканні; програма повинна мати також командний інтерфейс, який є подібним до архиваторів gzip, bzip2 або lzma.

Використовувані алгоритми: алгоритм Хаффмана.

СПИСОК ЛІТЕРАТУРИ

1 Игошин В.И. Математическая логика и теория алгоритмов : учеб. пособие для студ. высш. учеб. заведений / В. И. Игошин. – 2-е изд., стер. – М. : Издательский центр «Академия», 2008. – 448 с.

2 Игошин В.И. Задачи и упражнения по математической логике и теории алгоритмов / В. И. Игошин. – 3-е изд., стер. – М. : Издательский центр «Академия», 2007. – 304 с.

3 Романовский И.В. Вычислительная математика и структура алгоритмов. – М.: МГУ. 2006. – 112 с.

4 N. Wirth. Algorithms + Data Structures = Programs. Prentice-Hall, 1976.

5 A. V. Aho, J. E. Hopcroft, J. D. Ullman, Data Structures and Algorithms. Addison-Wesley, 1983.

6 Т. Н. Cormen; С. Е. Leiserson, R. L. Rivest, C. Stein. Introduction to Algorithms (3rd ed.). MIT Press and McGraw-Hill, 2009.

Навчальне видання

Стратієнко Наталія Костянтинівна

Шматко Олександр Віталійович

Бородіна Інна Олександрівна

**Методичні вказівки до виконання курсової роботи
по курсу «Алгоритми та структури даних»**

для студентів, які навчаються за напрямком
6.050103 «Програмна інженерія» спеціальності .05010301 «Програмне
забезпечення систем»

Відповідальний за випуск М.Д. Годлевський
Роботу до видання рекомендував О.В. Горілий

В авторській редакції

План 2016 р., поз. 79

Підписано до друку _____. Формат 60х84 1/16. Папір офсет.
Друк – ризографія. Гарнітура Times New Roman. Ум. друк. арк..
Наклад 50 прим. Зам № _____. Ціна договірна.

Видавничий центр НТУ «ХП». 61002, Харків, вул. Багалія, 21.
Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.

Друкарня НТУ «ХП». 61002, Харків, вул. Багалія, 21.